Astronomical Data Analysis Software and Systems VIII
ASP Conference Series, Vol. 172, 1999
D. M. Mehringer, R. L. Plante, and D. A. Roberts, eds.

DASHA-2: Improving Visualization and Processing of Photometric Data with IDL Objects

Oleg Smirnov

Institute of Astronomy of the Russian Academy of Sciences 48 Pyatnitskaya Str., Moscow 109017 Russia

Abstract. DASHA-2 is the next iteration of the DASHA package (Smirnov & Ipatov 1995) for processing data produced by the DAOPHOT II PSF photometry software (Stetson 1987, 1992). The original package was implemented under the PCIPS image processing system (Smirnov & Piskunov 1995), whereas DASHA-2 is completely written in IDL. The package is tailored for the reduction of large batches of CCD observations, where the same field is observed at different exposures and in different photometric bands. DASHA-2 is currently in heavy use at the Institute of Astronomy, where it is routinely employed to reduce observations of globular clusters, resulting in photometric data for upwards of 15,000 objects in a single field.

1. Introduction

DASHA-2 consists of four widget-based IDL applications, providing functionality for:

- Visualization of photometric data (e.g., photometric tables produced by DAOPHOT II – lists of stars detected in each CCD frame, along with their estimated PSF magnitudes.)
- Interactive filtering of photometric data using various goodness-of-fit statistics.
- Calibration of photometry using standards in the field.
- Cross-identification of objects between tables produced by DAOPHOT II for different CCD frames of the same field.
- Merging of data into a single master table, containing mean magnitudes for each detected object.
- Cross-identification of objects between master tables compiled for different photometric bands, and compilation of a single composite table containing magnitudes and color indexes.

The design of DASHA-2 made full use of the new object-oriented capabilities implemented in IDL[1] version 5. Development produced a powerful Foundation Class Layer (FCL), which allows the programmer to easily create new

[1]IDL is the trademark of Research Systems, Inc.

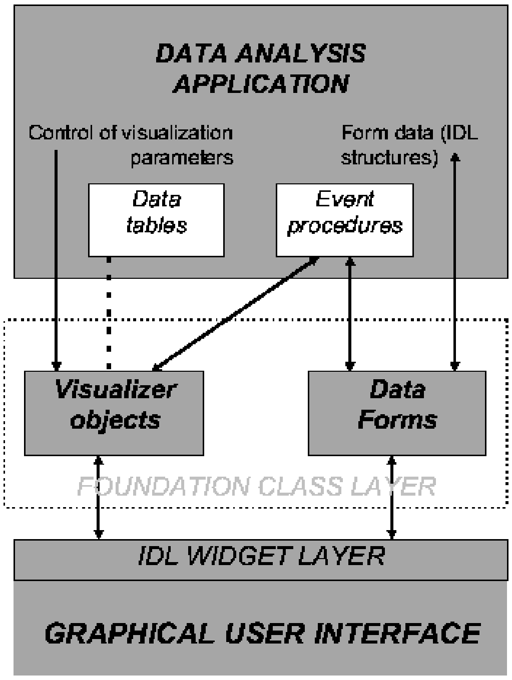

Figure 1. Structure of an application employing the FCL.

data analysis applications with sophisticated user interfaces and visualization capabilities.

The Foundation Class Layer hides the complexity of IDL widgets and graphics behind easy-to-use IDL classes. The two primary classes are the Visualizer, which provides a complex widget for interactive visualization of large data structures, and the DataForm, which simplifies the creation of complex user interfaces. This paper focuses on the benefits of the FCL for rapid development of scientific applications in IDL.

2. Structure of a DASHA-2 Application

The typical structure of an application employing the FCL is illustrated by Fig. 1. As one can see from the figure, the FCL completely isolates the GUI from the application. The rationale behind this design is as follows. With the ever-increasing amount of CPU power available to the average astronomer, interactivity is becoming the bottleneck for many data analysis applications. Whereas a decade ago the average user was mostly idle while the CPU was busy number crunching, today it is the average CPU that is mostly idle, waiting for the astronomer to prepare the data and evaluate the results. Thus, for many (though not all) data analysis applications, operational efficiency is determined more by a well-designed and functional GUI than by algorithmic efficiency. However, as any programmer can attest, GUI implementation is one of the more mundane and time-consuming tasks – and the more functional the GUI, the greater the pains required to make it all work. The FCL allows a developer to expend

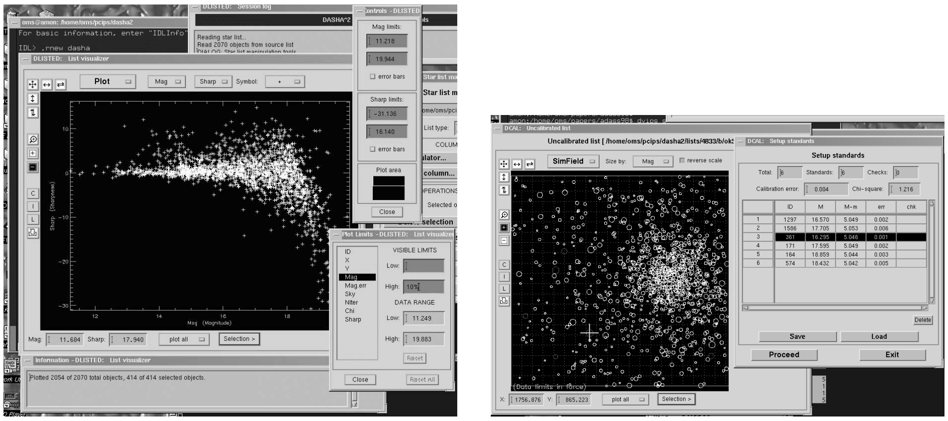

Figure 2. A Visualizer object (*left*). A sample DataForm object that can receive input from the Visualizer via an event procedure (*right*).

minimum effort on GUI implementation, and concentrate on the algorithmic aspect of an application. The FCL also ensures that all applications employing it have a similar look-and-feel (and allows improvements in the look-and-feel without touching the application code). While this concept is not new – it exists in many UI toolkits available today, the FCL is a very high-level toolkit specifically tailored to astronomical data analysis needs. For example, it provides powerful visualization capabilities, which can be very important for presenting results to the astronomer in an efficient way.

3. The Visualizer Object

Interactive visualization is available to any application, just by creating a Visualizer object and assigning it a data set. The visualizer object creates a complex widget-based GUI for visualizing the data set (Fig. Fig-D18-2, left), and then manages all user interaction transparently to the application. The Visualizer object encapsulates the following visualization functionality:

- Provides visualization (including hard copy) of any kind of data table (i.e., array of IDL structures).
- Allows the user to control visualizations on the fly, interactively generate plots, diagrams and histograms for any columns of the table, and control plot parameters.
- Zoom in and out of plot regions using the mouse pointer, track pointer position in data space.
- Set data limits on individual columns (as values or percentiles).
- Select data objects directly on the plot using click-and-drag, extend the selection across multiple plots, and filter displayed data based on the selection.
- Query objects by pointing to them on the plot.

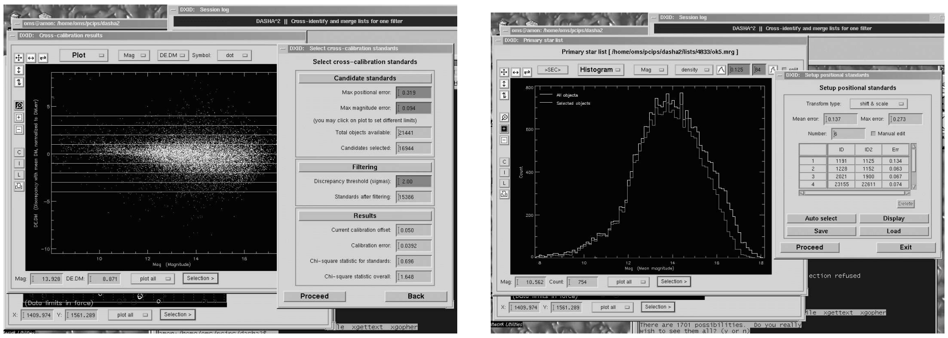

Figure 3. More examples of DataForms and Visualizers.

All functions of the Visualizer object may be controlled both by the user and the application. The application can interact with the Visualizer object to set up specific visualizations and receive user feedback.

4. DataForm Objects

DataForm objects allow an application to generate sophisticated user interfaces without the complexity of IDL widgets. Any application can create a DataForm object and fill it with text entry fields (which feature automatic input type checking), tables, and various standard controls (buttons, droplists, check-boxes, etc.), all in a few lines of code. Data exchange between the form and the application is simplified by mapping the form fields to an IDL structure; the form may be filled with initial values and later queried with a single call.

More sophistication can be added by employing event procedures. Event procedures allow forms to immediately respond to user input, and also handle events from Visualizer objects. The form in Fig. 2 (right), employed in the photometric calibration procedure of DASHA-2, features a table of photometric standards. The user can add standards to the table by simply selecting them on the Visualizer (here in "simulated field" mode) with the mouse. Other example DataForms and Visualizers are presented in Fig. 3.

5. Conclusion

While DASHA-2 is a complete package, the FCL is nowhere near that elusive state. For example, it still lacks a Visualizer object for 2D images and other data sets. Development of the FCL is continuing. It is planned to release an initial version to the public under a GPL-style license. For more information, please contact the author at `oms@inasan.rssi.ru`.

Acknowledgments. I would like to thank the LOC of the ADASS'98 Conference for the financial support that made this presentation possible. I would also like to acknowledge the Russian Academy of Sciences for the aid provided by the "Young Scientists Support Program".

References

Smirnov, O. & Ipatov, A. 1995, in ASP Conf. Ser., Vol. 77, Astronomical Data Analysis Software and Systems IV, ed. R. A. Shaw, H. E. Payne, & J. J. E. Hayes (San Francisco: ASP), 488

Smirnov, O. & Piskunov, N. 1995, in ASP Conf. Ser., Vol. 77, Astronomical Data Analysis Software and Systems IV, ed. R. A. Shaw, H. E. Payne, & J. J. E. Hayes (San Francisco: ASP), 133

Stetson, P. B. 1987, PASP, 99, 191

———, 1992, in ASP Conf. Ser., Vol. 25, Astronomical Data Analysis Software and Systems I, ed. D. M. Worrall, C. Biemesderfer, & J. Barnes (San Francisco: ASP), 297